IN THE SPECIFICATION:

Please replace paragraph 7, p. 3 with the following:

A1 The invention relates to automatic server-side plug-and-play without user intervention. An automatic plug-and-play component residing on the server is designed to detect connection and disconnection of a device to a port. The port may be, for example, the parallel port, a serial port, or another type of port. The device may be, for example, a printer, a scanner, a fax machine, or another type of device. Without user intervention, the component automatically installs an appropriate driver for the device upon connection of the device to the port. The device is then accessible by clients served by the server. That is, the device is shared among the clients. Similarly, without user intervention, the component automatically uninstalls the driver upon disconnection of the device from the port. The device is then inaccessible by the clients.

Please replace paragraph 21, p. 5 with the following:

A2 The port can be a parallel port, a serial port, or another type of port. Where the port is a serial port, it may have a Universal Serial Bus (USB) form factor, an IEEE 1394 form factor, or another type of form factor. The device 106 can be a printer, a scanner, a fax machine, a digital camera, a multi-function device (MFD) having printing, scanning, and/or faxing capabilities, or another type of device. The server 102 ~~can~~ itself may be a server appliance. A server appliance is a server that typically has a reduced set of functional capabilities and is designed for easy installation and maintenance. The server appliance may lack a dedicated keyboard, monitor, and/or pointing device, such that the appliance is accessed through a client connected to the same network as the appliance.

Please replace paragraph 23, p. 6 with the following:

A3 The server 102 includes a port driver 204 for the port to which the device 106 can be connected. The port driver 204 resides in the kernel mode 212. The port driver 204 is the low-level driver that passes signals from the server 102 to the device 106. It also passes signals from the device 106 to the server 102. The port driver 204 is not specific to the automatic plug-and-play of the invention, but rather is the low-level driver that is

A3 cont'd

used by any component within the server 102 to communicate with the device 106. A driver is generally ~~generally is~~ a device-specific control program that enables the server 102 to work with a particular device. Because the driver handles device-specific features, the server 102 is freed from the burden of having to understand and support the needs of individual hardware devices.

Please replace paragraph 24, p. 6 with the following:

A4

The server 102 has an automatic ~~play-and-play~~ plug-and-play component 202. The component 202 can be implemented as one or more software programs, objects, or other type of software modules. The component 202 embodies the plug-and-play functionality of the invention that has been described. The component 202 without user intervention automatically installs an appropriate driver for the device 106 upon connection of the device 106 to a port of the server 102. This results in the device 106 being accessible by clients communicatively coupled to the server 102. The component 202 also without user intervention automatically uninstalls ~~installs~~ the driver for the device 106 when the device 106 is disconnected from the port of the server 102. This results in the device 106 being inaccessible by the clients.

Please replace paragraph 38, p.10 with the following:

A5

The monitoring logic of the monitor 206 and the external monitor 306 starts at the third state 506. When a plug-and-play identifier has been detected, there is a transition 510 from the third state 506 to the second state 504. A driver corresponding to the plug-and-play identifier is installed. If a device driver is manually installed, then there is a transition 512 from the third state 506 to the fourth state 508. At the second state 504, if a new plug-and-play identifier is detected, then there is a transition 514 back to the second state 504, such that a new device driver, corresponding to the new identifier, is installed. Also from the second state 504, when the device is disconnected, the fourth state 508 is transitioned to ~~as~~ the transition 518, since an identifier is no longer detected. The second state 504 can transition to the first state 502, indicated by the transition 516, if the device driver is manually uninstalled. From the first state 502, when an identifier is

A5 Cont'd.

no longer detected, there is a transition 522 to the third state 506. Further, the first state 502 is transitioned to the second state, indicated by a transition 520, when a manual driver installation is detected. From the fourth state 508, the installed driver can be manually uninstalled to reach the third state 506 via a transition 524. Also from the fourth state 508, if an identifier is detected, then a driver corresponding to this identifier is installed, and the third state is transitioned to, via a transition 526.